Research paper

Digital certificates

Fiton Frangu [1], Fjollë Islamaj [2] and Gent Govori [3]

[1] fiton.frangu1@student.uni-pr.edu

[2] fjolle.islamaj@student.uni-pr.edu

[3] gent.govori1@student.uni-pr.edu

Received: 10 January 2021; Accepted: 10 January 2021;

Abstract: In transparent and untrustworthy environments such as the Internet, digital certificates are typically necessary for achieving safe communication. X.509 is the de facto digital certificate standard that is used on the Internet to construct the Public Key Infrastructure (PKI). Traditional X.509 certificates, however, are too heavy for Internet of Things (IOT) devices powered by batteries or energy harvesting, where it is important that energy usage and memory footprints are as minimal as possible.

In this paper, we present SmartCert, a new approach to the redesign and enhancement of digital certificate properties, which is a lightweight digital certificate for resource-limited IoT devices. Using the modern CBOR encoding method, we also suggest compression of the X.509 profiled fields. SmartCert is operated by smart contracts and SmartCert can provide the benefits of current PKI changes as well as new desired features and functionalities thanks to this technology. We also aim to cover not having protection mechanisms, but our protected approaches are just as critical as providing insight into what is happening with product data to help trust the data. We have an X.509 certificate usage analysis and review a proposal to incorporate X.509 digital certificates for product data authentication, authorization and traceability.

Keywords: X.509 Certificate; IoT; CBOR; PKI; SmartCert;

1. Introduction

Digital certificates are electronic credentials that connect the certificate owner's identity to a pair of (one public and one private) electronic encryption keys that can be used digitally to encrypt and sign information. The main purpose of the digital certificate is to ensure that the public key found in the certificate belongs to the individual to which the certificate has been issued, i.e. to verify that the person who sends a message is who he or she claims to be, and then to provide the recipient of the message with the means to encrypt the sender's reply.

To facilitate the distribution and recognition of public keys, encryption techniques using public and private keys require a public-key infrastructure (PKI)[1]. With either the public key or the private key, messages may be encrypted and then decrypted with the other key. One could send data encrypted with a private key without certificates and the public key would be used to decrypt the data, but there would be no guarantee that the data was originated directly by someone. All the user will know is that there was a legitimate key pair used. In essence, a Certificate Authority or CA is then a widely trusted third party to check the matching of public identification keys, e-mail, etc.

A digital certificate provides:

- Authentication, by serving as a credential to validate the identity of the entity that it is issued to.
- Encryption, for secure communication over insecure networks such as the Internet.
- Integrity of documents signed with the certificate so that they cannot be altered by a third party in transit.

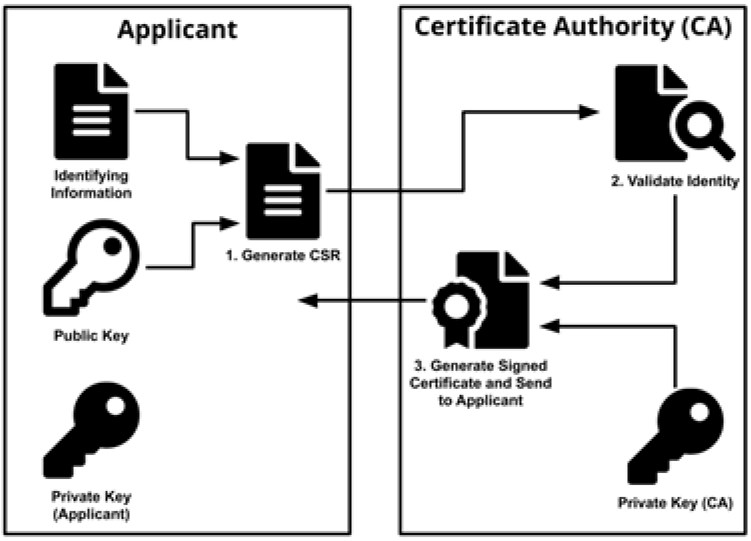

Figure 1 Public key certificate

A digital certificate applicant usually produces a key pair consisting of a private key and a public key, as seen on figure 1, along with a request for certificate signing (CSR). A CSR is an encoded text file containing the public key and other data to be included in the certificate (e.g. domain name, organization, email address, etc.)[2]. Usually, key pair and CSR generation are performed on the server or workstation where the certificate will be installed, and the type of information included in the CSR varies depending on the degree of validation and the certificate's intended usage. Unlike the public key, the private key of the applicant is kept secret and can never be revealed to the CA(or anyone else).

After the CSR is generated, the applicant sends the certificate to a CA, which independently verifies that the information it contains is right and, if so, digitally signs and sends the certificate to the applicant with the issuing private key. The receiver may cryptographically validate the digital signature of the CA through the public key of the CA when the signed certificate is shown to a third party (such as when that person accesses the website of the certificate-holder). In addition, the receiver can use the certificate to confirm that the signed material has been submitted by someone who holds the appropriate private key and that the information has not been altered since it was signed.

The use of digital certificates is defined by most IoT standards. Recently, we have shown that even though traditional X.509 certificates fit into state-of-the-art IoT hardware, they have considerable overhead on battery-powered IoT devices in terms of energy consumption. For workstations and servers in mind, traditional certificate requirements are established where factors such as computing capacity, memory footprint and energy usage are not primary concerns. These factors are critical, however, in battery driven and energy harvesting IoT devices, and it is therefore important to adjust these requirements to be more appropriate for IoT.

We used either pre-shared keys or regular X.509 certificates in the preceding work. Efforts have also been made to compress digital certificates without breaking the compatibility that uses traditional compression methods and dictionaries to compress X.509 certificates with repeated and commonly used text strings. The DEFLATE compression algorithm with a dictionary consisting of a typical certificate with unpopulated cryptographic fields is used in a modified version of gzip. These solutions are intended for traditional hosts on the Internet; they can, however, be used in combination with the solutions suggested in this paper[3]. A digital revolution has been initiated by developments in information technology such as big data, service-oriented architectures, and networking, promising lower costs, increased efficiency, and higher quality. Modern manufacturing firms are both more internationally dispersed and more digital than ever, leading to increasingly complex networks of manufacturing processes. Under these distributed manufacturing systems, manufacturers are under mounting pressure to conduct digital manufacturing more efficiently and effectively. Moreover, engineers are being pushed by industry and business demands to use more manufacturing information and knowledge in their design decisions . To do so, industry is changing how product definitions are communicated – from paper to models.

This paper investigates and proposes a lightweight implementation of a digital certificate with properties such as low memory footprint, low computational complexity and minimised data transfer as the main concerns. The solution proposed in this paper consists of two parts.

The first component is an IoT X.509 Profile that defines the appropriate field to be used without violating security and standard enforcement when interacting with IoT devices. The second part defines compression mechanisms for profiled X.509 certificate fields, which are implemented when a certificate moves within 6LoWPAN networks, in order to further minimize the size. The second part defines compression mechanisms for profiled X.509 certificate fields, which are implemented when a certificate moves within 6LoWPAN networks, in order to further minimize the size[4]. Certificates conforming to this profile will be fully legitimate X.509 certificates, and any organization capable of processing standard X.509 certificates will process them. However, without using the guidelines outlined in this paper, new IoT devices can not process the legacy X.509 certificates that are generated. Certificates for IoT devices have to be provided directly using this profile specification. New X509 lightweight certificates that adhere to this profile can be obtained by Legacy devices.

2. Background technologies

2.1 X.509 Certificates

For a long time, X.509 certificates have been around and are part of several protocols, such as Datagram TLS and IKEv2. The X.509 certificate consists essentially of three parts: I the subject matter information, the issuer and the certificate data, such as the serial number and validity dates; (ii) the subject's public key and its cryptographic algorithm; and (iii) the signature of the issuing certificate authority (CA) ('X.509'). For optional extensions, the new update (X.509 version 3) has been opened, which can be marked as essential and therefore must be processed by the recipient[5]. Using the Abstract Syntax Notation One (ASN.1) Distinguished Encoding Rules (DER), an X.509 certificate is specified and encoded and then converted to Base64 before being stored or transmitted.

2.2 CBOR and CDDL

A lightweight encoding scheme with support for binary data is Succinct Binary Object Representation (CBOR). In terms of code and message sizes, CBOR is designed to be extremely lightweight. CBOR is JSON based and completely follows the syntax and data types of JSON. In order to encode and decode messages, while CBOR does not rely on a particular schema, the CBOR Data Description Language (CDDL) was defined to define and restrict CBOR structures. We use CBOR to encrypt our X.509 profile certificate and finally compress it.

2.3 IoT Protocols: 6LoWPAN, CoAP, DTLS

Wireless Personal Area Networks (6LoWPAN) IPv6 over Low Power is a transmission protocol that allows IPv6 communication over low-powered and lossy networks, such as protocol IEEE 802.15.4. A Network Protocol (similar to HTTP) standardized for IoT is the Constrained Application Protocol (CoAP). Datagram TLS (DTLS) is mandated by Stable CoAP (CoAPs). CoAP has several limitations on how it is possible to build the certificates. The cipher fits and subject names to be used are defined in section 9.1.3.3 of the CoAP standard, which we take into consideration when designing an IoT certificate. For IoT x], the DTLS profile defines the use of the DTLS protocol in restricted environments. This profile also specifies the use of certificates and their contents, where restrictions have been made to keep the certificates smaller. Our work is in line with these guidelines.

2.4 Data authentication, authorization, and traceability

The Federal Aviation Administration (FAA) allows aerospace producers in the regulated U.S. aerospace industry to define a plan and obtain FAA approval for the management and maintenance of electronic design data (e.g., 3D computer-aided design (CAD) models, digital parts lists) used in the certification process. Then, in order to ensure compliance with relevant regulations, a parts manufacturer must be able to "[determine] the quality, eligibility, and traceability of aeronautical parts and materials intended for installation on U.S. type-certified products and items." This requires the manufacturer to know the correct type-certificated design data and if that data was used during production. Accomplishing this task is easier said than done. Today, the traceability process is often done with significant human capital and minimal-to-no automation.

It is also difficult to distinguish data traceability from authentication and authorization. Authentication is the act of deciding that an object is as declared by the entity (e.g., person, data)[6]. For example, to ensure a user is legitimate, the Public Key Infrastructure (X.509-PKI) is also used. Authorization, on the other hand, is the method of deciding which approvals are issued by a trusted source to an individual. Authorization methods, for instance, might specify how data can be used in a given process. In manufacturing, contracts between entities generally determine what information is declared to be and how data declarations are verified (i.e., authentication). However, authorization requirements are not negotiated typically such that a data user could know how data should be used during a prototype versus a production run.

For manufacturing industries, ensuring full data integration of authentication, authorization, and traceability is essential. These organizations need to be able to determine data statements, who did what, when they did it, and possibly why it was done with the data. For data authentication, authorization, and traceability, both regulated and non-regulated industries need effective and efficient processes. Significant resources are spent on data authentication, authorization, and traceability by controlled industries (e.g., aerospace, automotive, medical) to ensure they comply with the required public safety oversight[7]. Manufacturers in both regulated and non-regulated industries care about data authentication, authorization, and traceability to reduce product-liability exposure within their supply chains and in the public realm.

It is assumed that the cost of achieving data authorization and traceability outweighs the advantages of paper-based systems. Reports as far back as 2006 found that major original equipment manufacturers (OEMs) outsourced 60% to 80% of their manufacturing. The majority of OEMs today manufacture far less in-house goods, depending more on their external supply chains. For example, there are 30 tier-one suppliers for the Boeing 787 (Dreamliner), which in turn contracts with hundreds of tier-two and tier-three suppliers. In addition to the communication problems that come with drawing-based systems, monitoring what information is used by whom and for what reason for the Boeing 787 program is expensive and inefficient. In addition, it is a real issue to know and verify that the data being used is the actual FAA-approved data. This is why, using only 3D CAD models, Boeing chose to turn to an MBE to identify and certify the aircraft. However, for authentication, authorization, and traceability, 3D CAD models still lack commercial-off-the-shelf support. This is the reason for our research into the use of embedded X.509 digital certificates for authentication, and traceability of product data. authorization in 3D CAD models.

3. X.509 Profile for IoT

We suggest the X.509 certificate profile for IoT in this section and discuss individual fields and their mechanisms of compression. We also use the guidelines from the DTLS profile for the IoT standard in our design.[8]

Version. The new edition (since 2008) is 3, adding optional extensions. Version 3 is also the only legitimate version that is used for IoT in the DTLS profile. We also set the version value to 3 in our Profiled Certificate. Limiting the version number enables this field to be omitted when a certificate moves inside the 6LoWPAN network. The version field is set to 3 when a certificate leaves a 6LoWPAN network.

Serial Number. During the certificate enrollment process, a CA selects the serial number. We do not limit the value of the serial number; we recommend low numbers, however, and by encoding it in the CBOR format, the size is reduced. We also only use unsigned values based on the DTLS IoT profile guidelines.

Signature and signature Algorithm. The signature algorithm that a CA uses to sign a certificate is defined in these fields. The same value is found in both the signature and signatureAlgorithm fields. The signature and signature algorithm fields are omitted and the signature algorithm is set to ecdsaWithSHA256, which is used in the DTLS IoT profile as well. When the certificate leaves 6LoWPAN networks, this field is populated back to ecdsaWithSHA256 by the 6LoWPAN border router.

Issuer. To define the issuing CA, it is a non-empty sequence of name-value pairs that is used. Although the issuer is a key area for a certain CA to map a given credential, the range of possibilities for defining an issuer is vast and it is not appropriate for IoT devices to use the full range. This field is therefore limited to the common name (CN) of the UTF8String type. The name must not, however, be the same as that of any other known CA. Our CBOR coding of this field reduces the "Root CA" example from 20 bytes to 8 bytes.

Validity. It is a sequence of two dates which can be expressed in various formats: the start date and the end date. The ASN.1 representations used in conventional X.509 certificates, with up to six times more bytes available than UnixTime, are the longest of all. The text format is compressed to UnixTime, which allows the least number of bytes to represent a date: four bytes before January 2038 and five bytes after that. The structural specifiers are omitted because UTCTime is implied.

Subject. The subject field, similar to the Issuer field, represents the entity with the public key issued. Another CA or an end-user may be a subject. In both cases, the non-empty Distinguished Name must be (DN). The subject field in our profile, based on the DTLS IoT profile guidelines, contains the CN represented in the EUI-64 format when the certificate is issued to an IoT device. We reflect the CN in the format of the UTF8String used in the EUI-64 IEEE Guidelines. We compress the CN to the binary representation and CBOR format within 6LoWPAN networks, which takes the size from 36 bytes down to 9 bytes.

Subject Public Key Info. In a bit string, it includes the public key and the algorithm the key is used with. We fix the algorithm for our profiled certificate to the 256-bit ECC keys from the prime256v1 curve; therefore, when a certificate travels within 6LoWPAN networks, we omit algorithm details. Using the Miller method , we compress the ECC public keys.

Issuer Unique ID and Subject Unique ID. These fields are valid for version 2 or 3 only, and are only required if they are duplicated by the issuer or the subject. Subjects are fundamentally special in our solution, and issuers need to use unique names, which makes these fields redundant. Therefore, we omit these fields in the certificate profile.

Extensions. Three sections are composed of extensions; an OID, a boolean telling whether or not it is important, and an ASN.1 DER bit string encoded as the value. By omitting the first two bytes that will always be 0x551D, we compress the OIDs. For the CBOR layout, the remainder of the OID bytes are used as a tag that has the format: [tag, critical*, value], where critical is a true or false value and is the same as in ASN.1. As a compression mechanism for all possible extensions, the value will contain the DER-encoded bit string and its variants will be too complicated to fit into this basic protocol. In this profile, any extension is permitted. An example of the compressed extension field [[1, true, 0x3000], [15, 0x03020284]] is given here.

Signature. This is an encoded bit string which reflects a CA's actual digital signature. We use the format ECDSA-Sig-Value defined in RFC5480 . In an ECDSA signature, the r and s values are both 256 bits (32 bytes) unsigned integers, when the prime256v1 curve is used. R and s are not points on the curve, unlike the x and y values of an ECC public key, and can therefore not be compressed in the same way. However, proprietary solutions for ECDSA signature compression are available, such as compressed ECDSA signatures (patent number US 8631240 B2), where the s value is replaced by a lower c value. Within 6LoWPAN networks, we omit the signatureAlgorithm value (as it is fixed) and compress the signature by encoding it to the CBOR format, which reduces the size from 75 to 66 bytes.

Table 1 Summary of certificate field contents in the X.509 Profile for IoT

Field	Value
Version	3
Serial Number	Unsigned integer
Signature	ecdsaWithSHA256
Issuer	CommonName containing CA name as UTF8String
Validity	UTCTime in format YYMMDDhhmmssZ
Subject	CommonName containing CA name or EUI-64 as UTF8String
Subject public key info	ecPublicKey followed by prime256v1 and 64 byte uncompressed ECC public key
Issuer and subject unique ID	Not present
Extensions	Any extension
Signature algorithm	ecdsaWithSHA256
Signature	ECDSA-Sig-Value ::= SEQUENCE {r INTEGER, s INTEGER}

4. Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI) is a series of servers that are used for generating and maintaining digital certificates and public keys. It creates digital certificates that connect public keys to individuals, securely store them and, if necessary, revoke them. Passwords are typically used for authentication, for information transfer, but a more reliable authentication method is required to transfer sensitive information in a distributed environment. Such a method of authentication must confirm the identity of the individuals involved in the communication and the information being transmitted must be authenticated. The public-key infrastructure is one such strategy. So the public key infrastructure provides a secure environment for online transactions, confidential email and e-commerce by:

- Authenticating the identity of the entities (the sender and the receiver)
- Maintaining the data integrity

Consider an instance where, say Alice and Bob, asymmetric key encryption is used for communication between two individuals[9]. To encrypt the message, Alice uses Bob's public key and sends it to Bob. Using his private key, Bob decrypts the received message and reads the message. The only downside to this approach is that there is no way to guarantee that Bob or any other malicious person pretending to be Bob really owns the public key that Alice used for the encryption of the message. PKI has developed to solve this problem. PKI uses a trusted third party to distribute keys and to authenticate the identities to guarantee the identity of the individuals in a conversation. This is done by integrating digital certificates. Components of public key infrastructure are:

- Certificate Authorities
- Registration Authorities
- Certificate Repositories
- Digital Certificates

Advantages of using Digital Certificates in Public key Infrastructure:

1. Authentication: By using digital certificates the identity of the entity can be verified.
2. Secure: It assures that the public key belongs to the owner of the certificate and so a secure communication can be established for confidential email, e-commerce and online transactions.
3. Integrity: Integrity is guaranteed a long as the CA's signature on the digital certificate can be verified.
4. It prevents man-in-the-middle attacks, where a malicious user pretends to be the web site that the user wants to connect with.
5. Non-Repudiation: The signature on the certificate guarantees that only the web site owner has the private key associated with the public key in the certificate.
6. The process of verification and authentication is transparent to the end user and the process of authentication takes only a few milliseconds[19].
7. Certificates are supported by many enterprise networks and applications.

Disadvantages are**:**

1. A browser does not give a warning when a web site changes the certificate.
2. A user has to blindly trust that the developer of the OS has installed genuine root certificates and not fraudulent certificates.
3. A fraudulent root certificate can be installed in the browser when a malicious user gains access to the personal computer. In this case the browser will not report any security warning while browsing sites that use the fraudulent certificate.

4.1 Certificate Authority

A certificate authority (CA) is responsible for signing certificates in the X.509 confidence model. These certificates act as an introduction, which means that a CA acts as a trustworthy third party, between two parties. A CA processes requests from individuals or entities seeking certificates (called subscribers), verifies the information and, based on that information, theoretically signs an end-entity certificate. A CA needs to have one or more widely trusted root certificates or intermediate certificates and the corresponding private keys to play this function effectively. By getting their root certificates included in common applications, or obtaining a cross-signature from another CA delegating trust, CAs may accomplish this large trust. Within a relatively small society, such as a corporation, other CAs are trusted and are spread through other mechanisms such as Windows Group Policy.

Functions of the CA are:

- Issuing certificates
- Maintain and issue Certificate Revocation Lists (CRLs)
- Publish its certificates and CRls
- Maintain status information of certificate expiration dates

The CA can delegate these tasks. Certificate issuance is one of the key roles of a certificate authority (i.e. creating the certificates and signing them). Consider a server that has a digital certificate requested for itself[10]. When the registration authority has checked its identity, the request is then forwarded to the certificate authority. A certificate is generated in a standard format by the certificate authority (X.509 certificate standard).

The certificate includes the server's identity and its public key. This certificate is then signed with its own private key by the certificate authority, and the certificate is given to the requesting server. The signature of the CA on the certificate verifies the certificate's credibility. A copy of the certificate is saved locally and can be distributed in public repositories as well.

The Certificate Authority is the origin of the public key infrastructure trust. There is a root CA, which has its own digital credential, when a hierarchical CA architecture is followed. A certificate like that is self-signed. By signing the certificates of the subordinate certificate authorities, the root CA generates a chain of trust. This implies that the root CA trusts the certificates issued by subordinate CA's. So, if the root CA is trusted, a web browser or a user may trust the certificate provided by the subordinate CA. Most web browsers and operating systems have embedded the root CA certificate in them. For example in Internet Explorer, if we go to the Content tab in Internet Options, we can see the certificates. It has all certificates of the CA’s that the browser trusts.

The certificate authority's other role is to maintain and issue the Certificate Revocation List (CRLs)[11]. It is the responsibility of the certificate authority to cancel, remove or renew a certificate. It can be suspended by the CA during the lifetime of any certificate. Its validity is temporarily suspended at this point. Any time before its usual expiration, the CA can revoke a certificate. The certificate is not valid at this point. This can happen when a private key is lost or when knowledge of the private key is obtained by an unauthorized user. In such cases, the CA updates the CRLs and their internal records with the certificate and time stamp information needed. The CRLs are signed and put in a public repository by the CA. The Certificate Authority issues an expiry date for the certificates. It can no longer be used for authentication because the certificate has expired. The certificate owner is told of the upcoming expiry of the certificate in order to allow the customer to follow the process of renewal.

4.2 Registration Authority

The Public Key Infrastructure is part of the Registration Authority. It verifies digital certificate requests by validating the identity of the person making the request. For example, if a business demands a digital certificate, the RA verifies the owner's identity by checking different identity documents, such as a driver's license or a pay stub, etc. The RA then forwards the legitimate request to the CA after checking the identity. Then the CA issues a digital certificate. A CA is capable of having more than one RA. Each RA has a name and public key by which it can be recognized by the CA. Any RA is accredited by its respective CA. Any message that the CA receives with the RA’s signature is a trusted message.

4.3 Certificate Repositories

Certificate servers are generally used to store certificates and to distribute them. All the certificates given are deposited in the archive so that they can be easily accessed by the applications. For this method, a directory structure is best used. The Lightweight Directory Access Protocol (LDAP) is currently one of the best certificate repository technologies. These directories store certificates and allow the retrieval of these certificates for a user through applications. A large number of certificates are supported by this directory scheme. For such certificates, it stores the certificates and the public keys. The key benefit of these directories is that they can be used and made freely available on massively distributed networks. It also makes the search simpler in a hierarchical system by storing the certificates. The certificate registry also provides information about certificate status and revocation information. It also updates the certificates and their status, apart from storing and distributing them.

5. Web of Trust

The network of trust is an alternative to the centralized chain of trust certificate authority model of PKI. In the PGP 2.0 manual, the idea of the network of trust was first defined in 1992: "As time goes on, you will accumulate keys from other individuals that you may want to designate as trusted initiators." Everyone else is going to choose their own trusted initiators. And with their key, everyone will eventually accumulate and distribute a list of certifying signatures from other individuals, with the hope that at least one or two of the signatures will be trusted by those receiving it. This will lead to the development of a decentralized network of fault-tolerant trust for all public keys[12]. There are no certificate authorities in a web of confidence. Instead, any user of the system may sign the public key of each other, which ensures there is no "too big to fail" certificate authority in the system as public keys are built to have many signatures by default. This implies that the effect on the trust network is minimal if one signer is compromised and the signer's key is revoked.

6. Compressed X.509 Format (CXF)

There is a good attempt to compress X.509 certificates when it comes to digital certificates, without breaking the compatibility. Pritikin et al. used conventional compression methods and dictionaries to compress X.509 certificates with repeated and commonly used text strings. The DEFLATE compression algorithm with a dictionary consisting of a typical certificate with unpopulated cryptographic fields is used in a modified version of gzip. The results show that the X.509 certificates used as a test could be compressed at a rate of 0,86 for the RSA certificate and 0,73 for the ECC certificate, determined by dividing the compressed size by the original size. The unique 753-byte RSA certificate was split into 557 bytes of cryptographic data, 43 bytes of text and numbers, 42 bytes of entity identifiers (OIDs) and 111 bytes of structural details. These findings are remarkable, considering the incompressible nature of the cryptographic data, which is 0.74 of the total size of the certificate. By creating a new dictionary from a sample of 100,000 certificates, blogger Graham Edgecombe has recently expanded the CXF certificate[13]. Compared to the CXF dictionary, this dictionary turned out to minimize the size of the certificates by an extra 14 percent. The results provided by Edgecombe differ somewhat from the CXF specification experiments, with an even greater compression rate. The main explanation for this could be that Edgecombe uses actual authentic certificates, while one self-made dummy certificate is used by the CXF specification experiment. This reality makes the CXF strategy much more feasible as it is the success of real life that is of interest.

6.1 The CXF Dictionary

DEFLATE compression algorithm allows the use of a "preset dictionary" to make compression more efficacious for particular applications; it especially benefits the compression of shorter inputs. When a preset dictionary is used in the compression algorithm, the dictionary is fed into the compressor, and no output is produced while the dictionary is processed, but the compressor state is updated and maintained; after that, the data input is compressed. Decompression with a preset dictionary works similarly; data that is compressed with a particular preset dictionary must be decompressed with the same preset dictionary, or the output of the decompressor will not match the input of the compressor.

7. The Blockchain Paradigm and the Bitcoin System

In the Bitcoin electronic cash system , the idea of the blockchain was first implemented. Bitcoin is structured as a peer-to-peer network where transactions are relayed to other nodes by nodes running the Bitcoin program. "The network reaches a consensus on the ordering of transactions by recording them on the blockchain to prevent cash from being double spent; the Bitcoin paper describes a process of time stamping transactions by "hashing them into an ongoing hash-based proof-of-work chain, creating a record that can not be altered without redoing the proof-of-work. The proof-of-work in the Bitcoin network involves hashing blocks of incoming transactions repeatedly before a hash that is below a certain value is found. This requires computing power, and as long as the majority of the processing power on the network is not regulated by a central authority, the blockchain and reverse transactions cannot be changed by the central authority[14]. This is because the network's consensus rule is such that the blockchain with the most proof of-work is the right one, and so it is impossible that an authority that does not have the majority of the computing power of the network can outpace the production of blocks of the rest of the network. The accessible nature of the Bitcoin blockchain and the fact that it is built to render the central authority's control of the blockchain and reverse transactions computationally costly (and therefore potentially economically unattractive) allows it a useful tool to create decentralized and transparent applications where records can not be concealed or manipulated by a third party. This is particularly useful in the context of building a transparent PKI, as rogue certificates or identities would be universally visible.

The Blockchain is increasing, by design, every ten minutes at an average rate of one new block. After the genesis block, anyone can run a node of the Bitcoin network and maintain a copy of all the blocks and all the transactions used in them. In addition, anyone can send a transaction to other nodes, and if the transaction is legitimate, the miners will include it in the next block. When a small number of new blocks follow the block, including the transaction, the transaction can safely be called "permanently stored" in the Blockchain[15]. The Blockchain can thus be used as a publicly available, write-only, and time-stamped storage medium by properly utilizing the limited space for a message permitted in Bitcoin transactions. This is what makes it an efficient digital certificate network.

7.1 Blockchain discussion

Blockchain-based ventures have been a hot topic over the past two years. Most of the rhetoric is likely to have been overstated, and the same is true for some of the criticism. It would be a mistake to assume that we are dealing with an immediate implementation technology, or that improvements can be quickly enforced, since the Blockchain is a very modern and complicated technology. Rather, we are in an initial phase of exploration. The Blockchain is not the answer for today's credentials that can address anything that is incorrect. It does, however, offer some possibilities to enhance the structure we have today, and that is why exploring it is so fascinating and challenging. Blockchain ventures have raised alarm about the fervor with which certain industries have been pursuing the implications that a literal implementation of this technology could have. Talking about the ramifications of potential implementations at this time means going as much into the realm of speculation as the literature that promotes it.

Blockchain is a revolutionary technology that reveals itself to be an open resource with possibilities in various fields after a few years of implementation as the basis of the digital currency. The key to the interest in this technology lies in its ability to switch from a centralized data logging system to a distributed system that guarantees that the information is not changed and that privacy is protected. Blockchain's true breakthrough is that "confidence" is redefined as "high-trust computing" since you no longer have to trust anyone but an algorithm. It offers all kinds of data exchanges with reliability, accountability and security: financial transactions, contractual and legal agreements, changes of ownership, and certifications. For the business and the public sector, the blockchain offers an unparalleled opportunity.

Any institution that can take advantage of these technologies will have the opportunity to radically streamline and improve existing processes, create entirely new business models and develop innovative products and services for the new generation of consumers. However, this is not a vision of a utopian, tech-enabled future: technological capabilities are available today to keep an unalterable record of every exchange, eliminating the need for trusted third-party intermediaries in digital transactions. Faster operations, real-time visibility of transactions and reduced costs in every manufacturing, social and economic field are the result. Gartner estimates that by 2025, Blockchain will generate value-added revenue of US$ 176 billion, revolutionize the supply chain, allow new business models, and disrupt existing ones. In protecting knowledge archives and being a complementary element of information governance, blockchain technology may also play a significant role. Indeed, a great deal of literature has been based on big data analytics for decision-making, but more research on security and clearance permissions in private settings is needed. In filling this void, Blockchain may help to demonstrate other possible applications and uses of this technology. In addition, a useful and important topic to be deepened in future research is the collaborative essence of big data analytics with Blockchain technology.

One of the most commonly asked questions was why they chose Bitcoin Blockchain and not others, such as Ethereum, as MIT and Learning Machine were working on their Blockchain certificate project. The response was that Ethereum was only just at the beginning of an idea when MIT began the project, while Bitcoin was the most tested and consistent Blockchain to rely on. Furthermore, the relatively strong self-interest of miners and the financial investment made in Bitcoin (and Bitcoin-related companies) make it possible to be used for a long time.

8. Redesigning Digital Certificates with Smart Contracts

SmartCert, a novel approach to redesigning and improving digital certificate assets. SmartCert is operated by smart contracts and SmartCert can provide the benefits of current PKI changes as well as new desired features and functionalities thanks to this technology. A SmartCert certificate offers more comprehensive details about its state of validity, which is constantly evolving, but only with regard to the smart contract code defined and individual domain policies. Certificates provided and revised by CAs are kept accountable and their acts are clear and code-monitored. We also present SmartCert implementation and assessment, and our results indicate that the system is efficient and deployable as today.

8.1 SmartCert Model

As in the TLS PKI, SmartCert introduces the same parties. Domain delivers a service via the TLS protocol (e.g., HTTPS). We believe that DNS domain names are identified by services, and servers hosting these services. (We use the terms "domain" and "server" interchangeably for a concise description.) We believe that domain servers will communicate with the blockchain network, obtaining MTP proof. CA is a trusted entity that certifies that identities and their public keys are binding. We believe that CAs have key pairs that can be used around the platform of the blockchain, and CAs can send transactions to the platform of the blockchain[16]. CAs are similar to TLS PKI CAs in SmartCert, except that they are not fully trusted and their acts are self-monitored and self-imposed by a smart contract code and domain policies. The client wishes to contact a provider in a safe manner (served by a domain).

Clients trust CAs and they are also able to obtain the blockchain block headers from blockchain light clients. We believe that these parties will communicate with a smart contracts-supporting blockchain platform. We focus on Ethereum in particular, but SmartCert can be adapted to most smart contract platforms available. We assume various adversary models for demonstration, and first, an adversary of MitM is assumed. We presume that MitM attacks are short-lived; otherwise, any domain-validation scheme will be rendered ineffective by a permanent attack. The adversary's aim is to execute an undetected impersonation attack. We presume that the cryptographic methods used are reliable and that the opponent will not compromise the security properties of the blockchain network underlying it. In addition, we believe that CAs may misbehave by not performing (or wrongly performing) their duties, but all such misbehavior should be observable. We also consider a stronger opponent who can clash with m malicious CAs to impersonate a domain; however, we believe that the domain can still prove its identity to n honest CAs, for n > m.

8.2 Policy Contract

It is a smart contract that regulates the security policies of domains. Policies define when certificates for domains are considered legitimate. We assume that, with a publicly known address, there is one global example of this contract. Each domain should have a single policy at a time and the default policy is used if there is no domain policy.

8.3 SmartCert contract

It is a smart contract that incorporates, encodes, and enforces the logic and regulation enforcement of public-key validation. The logic of validation is represented as a code (in the programming language of the blockchain platform) with associated storage. The contract modifies its internal storage, which represents the validation state and its compliance with the domain regulation, by executing the logic (accessed from the policy contract). Arbitrary logic can be enforced through SmartCert contracts, but we assume that only such structured validation logic would be used in practice.

8.4 SmartCert certificate

Is an information that provides TLS customers with the current validation state of the contract in a stable and efficient manner. Clients accept or deny connections to TLS based on provided SmartCert certificates. In a setting where SmartCert certificates carry information about historical public-key validations of a given domain performed by multiple CAs, we present SmartCert throughout the paper. However, with almost arbitrary validation logic, SmartCert can be applied as it is only constrained by the deployed language of the smart contract.

8.5 SmartCert Security Analysis

SmartCert offers various security advantages over traditional certificates. First of all, SmartCert offers a reliable and clear, but strong, policy framework that promotes domain expressiveness. Domains may publish their policies to determine the conditions under which they consider SmartCert certificates to be legitimate. Multiple CAs (at least one) will authenticate policies; SmartCert is therefore no longer a weaker-link device. In addition, if a domain policy is released, it ensures that a) no SmartCert contract (and, subsequently, no SmartCert certificate) can be produced without the permission of the domain, and b) the policy would be self-enforced by all SmartCert contracts claiming the domain name.

For the former, if a domain name policy exists, a SmartCert contract with that specified name can be created only if it is authenticated by the policy key of the domain.

For the above, SmartCert contracts verify their compliance with the regulation for each update; hence, any non-compliant contract will invalidate itself with the first update and will therefore be rejected by customers (note, that clients check if the SmartCert contract code is correct to ensure that the validation rules are respected).

In addition to multiple authority authentication policies, valid SmartCert contracts must be successfully authenticated, as defined by the domain, by a minimum MIN CA CA from the CA collection. Therefore, it enhances the protection as several authorized CAs validate certificates and restricts the capabilities of the adversary as an adversary that exploits a CA can only update certificates that have authorized this particular CA. By changing its policy, an adversary capable of compromising multiple (m) CAs may try to impersonate a domain. Such an attack would be effective if the number of CAs signed by the current legitimate policy is at least equal to m. However, as assumed the domain can obtain a new policy signed by n > m CAs, and submit it to invalidate the malicious policy.

SmartCert contracts encode their existing validation states; it is therefore easy to get a clearer view of the contact domain public keys. More precisely, a SmartCert contract encodes the primary continuity measure in the presented setting and verifies its compliance with the domain policy, and then represents compliance with the SmartCert certificate. CAs are obliged to regularly perform public-key validations, and the code of the smart contract enforces and tracks this process. Such a design makes it possible to provide notary systems with the capability of detecting and preventing various MitM attacks where, for example, an opponent impersonates a website displaying a different certificate. If the malicious key is presented to a CA, it will be clear as it will be reflected in the SmartCert contract. In addition, CAs can not prefetch validation proof without breaking the blockchain network as they require recent pseudorandom block hashes. SmartCert contracts are structured in such a way that only approved CAs can update them, and the code and individual domain policies validate such updates. SmartCert therefore minimizes trust put in CAs[17].

If a malicious or inaccessible CA does not submit within an epoch a validation proof or submits an invalid validation proof, it will be detected and recorded in its internal storage by the SmartCert contract. Similarly, the SmartCert contract will not accept replayed or expired validation evidence as true and an error will also be registered. SmartCert therefore provides an essential property that will be reflected in the corresponding SmartCert contract if a domain is under a MitM attack during an epoch, and the issuing CA can not deny or conceal this truth. This model is useful not just for TLS users, but also for domains that can easily detect attacks just by looking at their SmartCert contracts.

SmartCert profits in other respects from the underlying blockchain network. High availability, which turned out to be an Achilles heel of almost all safety infrastructures, is an immediate advantage. SmartCert certificates are produced and distributed via the blockchain platform and contain all the necessary information to validate them (like a validation state and a revocation status). Just by interacting with the blockchain network, which is distributed, open, and resilient to censorship, domains obtain new MPT proof. CAs only need to be able to send transactions to the blockchain network, so to support TLS clients, they do not need to invest in any highly accessible front-end servers. It dramatically enhances the system's security, as an opponent attempting to launch a Denial-of-Service attack on a CA or trying to censor CA transactions needs to compromise the blockchain platform's assets, which requires (significant) control over the blockchain network. Transparency is another major advantage of basing SmartCert on the blockchain network. Publicly available are all regulations, licenses, and other CA actions.

8.6 Limitations

Although SmartCert offers several advantages, there are some limitations. First, we are mindful that our framework introduces changes to the existing environment that could be seen as radical, despite developing SmartCert with deployability in mind. We show that even when designed on existing resources and platforms, the system is feasible and effective, but given deployment experiences with other TLS PKI improvements, we should not presume that SmartCert can be deployed immediately as it is[18]. Second, the underlying blockchain platform affects various facets of SmartCert (such as economics, confidence assumptions, and throughput). In an open environment, we tested SmartCert to demonstrate that the cost of the certificate is fair, even in such a difficult implementation scenario. We do not see any reason, however, why SmartCert could not be deployed with a more powerful approved blockchain (e.g., run by a consortium of CAs, browser vendors, and non-profit organizations). Alternatively, SmartCert, such as Facebook's Libra, could also be applied on top of incoming approved networks, which would reduce the cost of launching and maintaining a new platform. Finally, the adoption of TLS is increasingly growing through projects providing free certificates. We see it as a very positive trend, but we do not believe that SmartCert can compete with (free) DV certificates in adoption as the cost of SmartCert deployment may impede domains that are happy with the current level of security. We see our method, instead, as a safer alternative to security-savvy domains. We claim that certain domains have no choice but to deploy domain-validated certificates in the face of the abandonment of EV certificates.

9. Conclusions

Digital certificates help overcome the security limitations of digital signatures by identifying the owner of the public key and making it available to all parties who need to validate it. Today, there are several different types of digital certificates, and they all play an important role in any comprehensive cybersecurity strategy.

This paper proposes SmartCert, a novel system for digital certificates. SmartCert implements "dynamic" certificates by leveraging the blockchain and smart contract technologies, whose states can change but only according to the encoded validation logic and security policies defined individually by domains. Thanks to this layout, SmartCert offers functionality and characteristics that previous systems do not provide. SmartCert certificates self-enforce versatile security policies, bear historical statistics of public-key validations performed, reduce confidence put in CAs as their activities are transparent and code-monitored, and provide high availability, robustness, and resistance to censorship. SmartCert needs no substantial changes to the PKI of the TLS and is paired with the TLS protocol. The framework is functional, does not add large overheads and, co-existing with X.509v3 certificates, can be implemented incrementally. In the sense of public-key validation, we presented SmartCert; but in the future, we will explore other validation policies. The SmartCert system, for example, may be expanded to support more complex security policies that allow domains to inform the client about other connection attributes (e.g. the desired recipher suites for a domain).

10. References

[1] What Is a Certificate Authority (CA)? Available online: https://www.ssl.com/faqs/what-is-a-certificate-authority (accessed on 08 Jan 21).

[2] Basics of Digital Certificates and Certificate Authority. Available online: https://sites.google.com/site/ddmwsst/digital-certificates (accessed on 09 Jan 21).

[3] Deflate https://en.wikipedia.org/wiki/Deflate (accessed on 03 Jan 21).

[4] Securing the Internet of Things: A Standardization Perspective. Available online:https://www.researchgate.net/publication/263128758_Securing_the_Internet_of_Things_A_Standardization_Perspective (accessed on 04 Jan 21).

[5] Illustrated X.509 Certificate. Available online: https://darutk.medium.com/illustrated-x-509-certificate-84aece2c5c2e (accessed on 07 Jan 21).

[6] Embedding X.509 Digital Certificates in Three-Dimensional Models for Authentication, Authorization, and Traceability of Product Data. Available online: https://www.nist.gov/publications/embedding-x509-digital-certificates-three-dimensional-models-authentication (accessed on 09 Jan 21).

[7] PKI4IoT: Towards public key infrastructure for the Internet of Things. Available online: https://www.sciencedirect.com/science/article/pii/S0167404819302019 (accessed on 03 Jan 21).

[8] Giancarlo Fortino; Carlos E. Palau; Interoperability, Safety and Security in IoT; Publisher: Springer International Publishing 2017, pg 125-130

[9] Public key infrastructure. Available online: https://en.wikipedia.org/wiki/Public_key_infrastructure (accessed on 06 Jan 21).

[10] How Does PKI Work? Available online: https://www.venafi.com/education-center/pki/how-does-pki-work (accessed on 05 Jan 21).

[11] Certificate Authority. Available online: https://www.sciencedirect.com/topics/computer-science/certificate-authority (accessed on 10 Jan 21).

[12] FAQ: Web of Trust. Available online: https://www.thenation.com/web-trust/ (accessed on 08 Jan 21).

[13] Digital Certificated for IoT. Available online: https://kth.diva-portal.org/smash/get/diva2:1153958/FULLTEXT01.pdf (accessed on 08 Jan 21).

[14] Blockchain Technology Explained. Available online: https://www.leewayhertz.com/blockchain-technology-explained/ (accessed on 07 Jan 21).

[15] Everything you need to know about Blockchain Technology. Available online: https://www.investopedia.com/terms/b/blockchain.asp (accessed on 04 Jan 21).

[16] Digital Certificates with Smart Contracts. Available online: https://www.researchgate.net/publication/340294956_SmartCert_Redesigning_Digital_Certificates_with_Smart_Contracts (accessed on 09 Jan 21).

[17] Digital Certificates. Available online: https://www.azion.com/en/documentation/products/edge-application/digital-certificates/ (accessed on 09 Jan 21).

[18] Blockchain and Smart-contract: a pioneering Approach of inter-firms Relationships? https://halshs.archives-ouvertes.fr/halshs-02111603/document (accessed on 08 Jan 21).

[19] Advantages of Public Key Technology https://www.safelayer.com/en/resources/59-articles/public-key-infrastructure/421-advantages-of-public-key-technology (accessed on 07 Jan 21).